BIDIRECTIONAL NETWORK-ON-CHIP ROUTER IMPLEMENTATION USING VHDL

Thesis submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

In

Electronics and Communication Engineering

By

Nupur Sahu (111EC0188)

Department of Electronics & Communication Engineering

National Institute of Technology, Rourkela

Rourkela - 769008, Odisha, India

May 2015

BIDIRECTIONAL NETWORK-ON-CHIP ROUTER IMPLEMENTATION USING VHDL

Thesis submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

In

Electronics and Communication Engineering

By

Nupur Sahu (111EC0188)

Under the guidance of

Prof. Ayas Kanta Swain

Department of Electronics & Communication Engineering

National Institute of Technology, Rourkela

Rourkela - 769008, Odisha, India

May 2015

CERTIFICATE

This is to certify that the thesis entitled **Bidirectional Network-on-chip Router Implementation using VHDL** by **Nupur Sahu** is a record of original research work carried out under my supervision and guidance for the partial fulfillment of the requirements for the degree of **Bachelor in Technology** in the department of **Electronics & Communication Engineering, National Institute of Technology, Rourkela**.

Place: NIT Rourkela

Date: May 10, 2015

Prof. Ayas Kanta Swain

Professor, ECE Department

NIT, Rourkela, Odisha

ACKNOWLEDGEMENT

Firstly, I would like to express my sincere gratitude towards my project supervisor and guide **Prof. Ayas Kanta Swain** for his advice and guidance during my project work. He has continuously encouraged me to stay focused on achieving my goal.

I would like to extend my thanks to our Head of Department, **Prof. Kamalakanta Mahapatra** and to all the professors of the department for their support.

Last, but not the least, I would like to acknowledge the motivation, love and support that I received from my parents and friends, and therefore I dedicate this thesis to my loved ones.

Nupur Sahu(111ec0188)

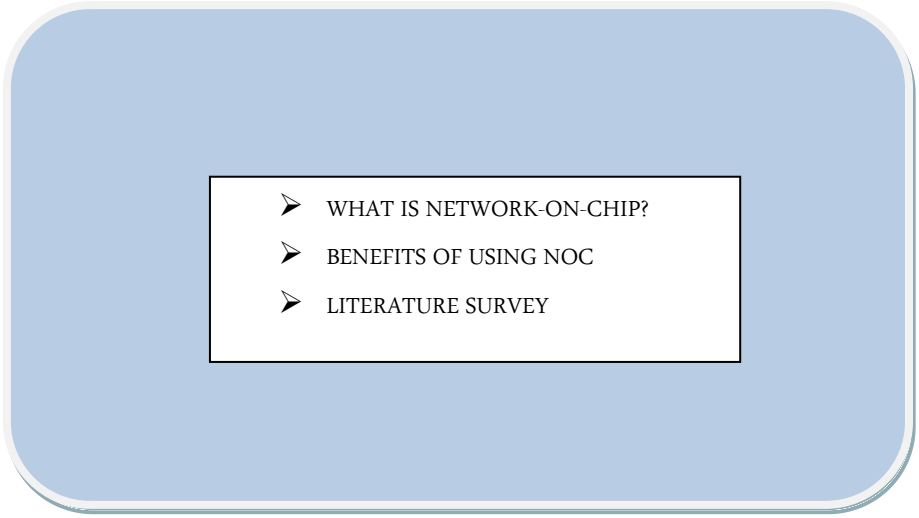
ABSTRACT

Network-on-chip (abbreviated as NoC) is emerging as a very reasonable and efficient answer for connecting the numerous cores in a System-on-Chip (abbreviated as SoC). Neighboring routers are connected through unidirectional communication channels in conventional Network-on-Chip architectures. Due to the uneven and unpredictable nature of traffic patterns in NoC, in one channel overflow might occur due to intense traffic in one direction but the other unidirectional channel is idle, hence causing performance degradation, loss of data and incompetent resource utilization. Bidirectional NoC (BiNoC) is used as a cure for this problem. It uses bidirectional channels to connect neighboring routers and supports recomputation of channel direction according to traffic demand during runtime by using CDC (channel direction control) protocol. Objective of this project is to design a packet switched Bidirectional NoC (BiNoC) having features like wormhole flow control, XY routing algorithm and 2-D mesh topology. The design architecture is coded using VHDL and simulated using Xilinx ISE tools.

Contents

TABLE OF FIGURES:

1. INTRODUCTION



1.1 WHAT IS A NETWORK-ON-CHIP?

According to Wikipedia:

"Network-on-chip (NoC) is a new exemplary for System-on-Chip (SoC) design. NoC based-systems accommodate multiple asynchronous clocking that many of today's complex SoC designs use. This solution brings a networking scheme to on-chip communications and claims roughly a threefold performance increase over conventional bus systems."

NoC is a communication system on an IC, generally IP cores in a System- on-chip (SoC). NoC uses networking theory to bring visible advantages over conventional crossbar interconnects and bus. NoC increases the scalability factor and power efficiency of SoCs.

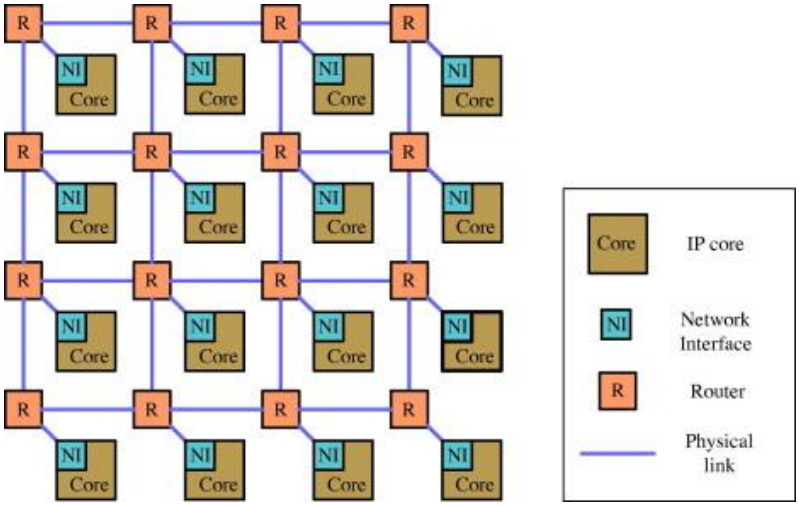


Figure 1: General Structure of SoC

Conventionally, System-on-Chips (SoCs) make use of topologies based on shared buses. Going by NoC design steps, designers use network design technology to design and analyze SoCs. NOC can be defined as 'network-on-chip, is a communication network aimed for on-chip'. NoC was proposed as an answer to multifaceted communication problems that occur on the chip. In NoC architectural style, the chip is broken into a set of interconnected nodes where each node can be a processor and is commonly known as processing element (PE). A router is present inside each node. It connects every node to its neighboring nodes. The router has four ports (North, South, East, West) for connecting with other routers and a local port for connecting with PE. Router basically consists of Routing algorithm and Switching techniques, where routing

algorithm proposes a selection procedure based on the idea of traffic-on-path. The routing algorithm can be used to avoid faulty ports. Switching techniques help determine how and when internal switches connect their inputs with the outputs and the time at which message components may be transmitted along these paths. Overall performance of a NoC depends on factors like flow control, switching techniques, topology, and routing algorithm.

1.2 BENEFITS OF USING NOC

The connecting wires present in the links of a Network-on-Chip are shared between a number of signals. A high degree of parallelism is gained here, because every link is able to work concurrently on separate data packets. So, as the complexity of an integrated system grows, a NoC provides improved scalability and performance compared to conventional communication architectures. Obviously, the algorithms should be devised in a manner that they offer huge parallelism and will thus be able to use NoC's potential to its maximum.

ICs are designed with point-to-point connections that are dedicated in nature, i.e. one wire for each signal. Particularly for bigger designs, this can have many limitations from the physical design point-of-view. The wires consume maximum of area of the chip. In nanometre CMOS, interconnects dominate dynamic power dissipation and performance.

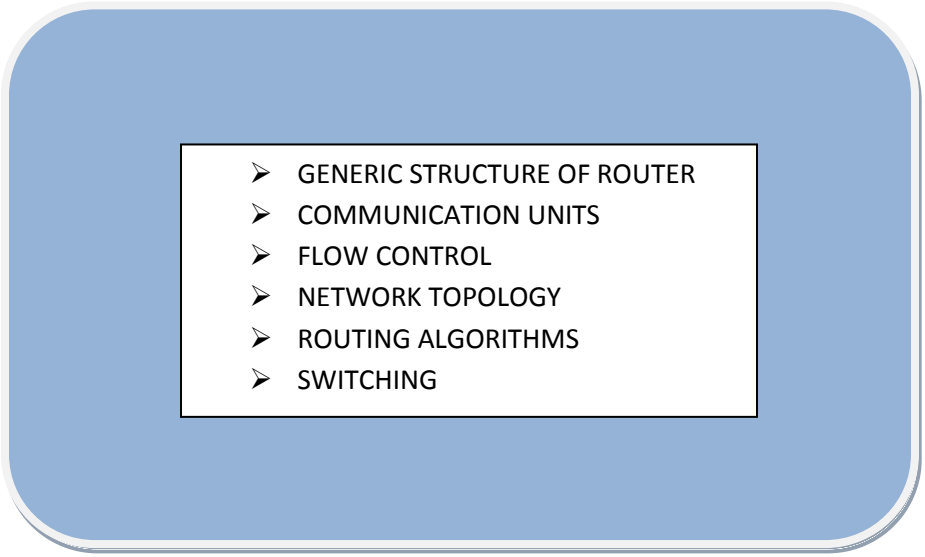
1.3 LITERATURE SURVEY

[1] Natalie Enright Jerger and Li-Shiuan Peh, "On-ChipNetworks" talks about Network on chip design starting from scratch. Its evolution, its components, architecture, working, etc with examples.

[2] Arun Arjunan, Karthika Manilal, " Noise Tolerant and Faster On Chip Communication Using Binoc Model", International Journal of Modern Engineering Research (IJMER) Vol. 3, Issue. 5, Sep - Oct. 2013 pp-3188-3195 ISSN: 2249-6645 talks about a Network-on-chip architecture with Error correction and retransmission schemes.

[3] http://pages.cs.wisc.edu/~tvrdik/7/html/Section7.html Tells us about communication units, terminologies, various flow control techniques and topologies used in NoC.

2. GENERAL TERMINOLOGY USED



2.1 ROUTING ARCHITECTURE:

Processing Node:
It is a standard bus-based computer provided with a router.

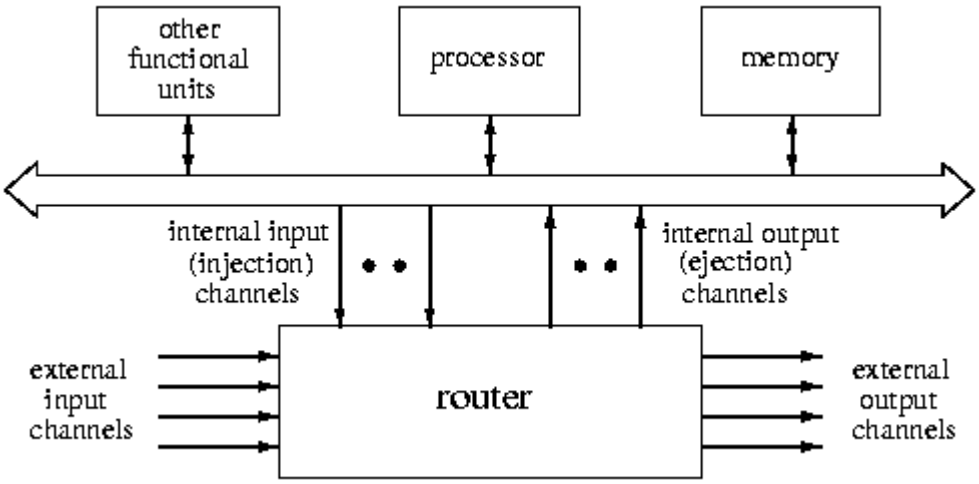


Figure 2: Structure of a Processing Node

Router:

It is a hardware co-processor that implements bottom levels of a communication protocol. It consists of a routing and arbitration unit, switch and memory buffers.

Adjacent nodes:

These are nodes with routers that are directly connected.

External channels:

They allow routers to connect among themselves and hence define the topology of a interconnection network.

Internal channels:

They establish connection between the local processing core with the router and implement the physical interface between them. They have 3 types of architectures:

- **1-port** : It consists of one ejection and one injection channel,
- ***K*-port** : It has *k* ejection and *k* injection channels,

- **All-port** : Here, number of external output channels equals the number of injection channels and number of external input channels equals the number of ejection channels.

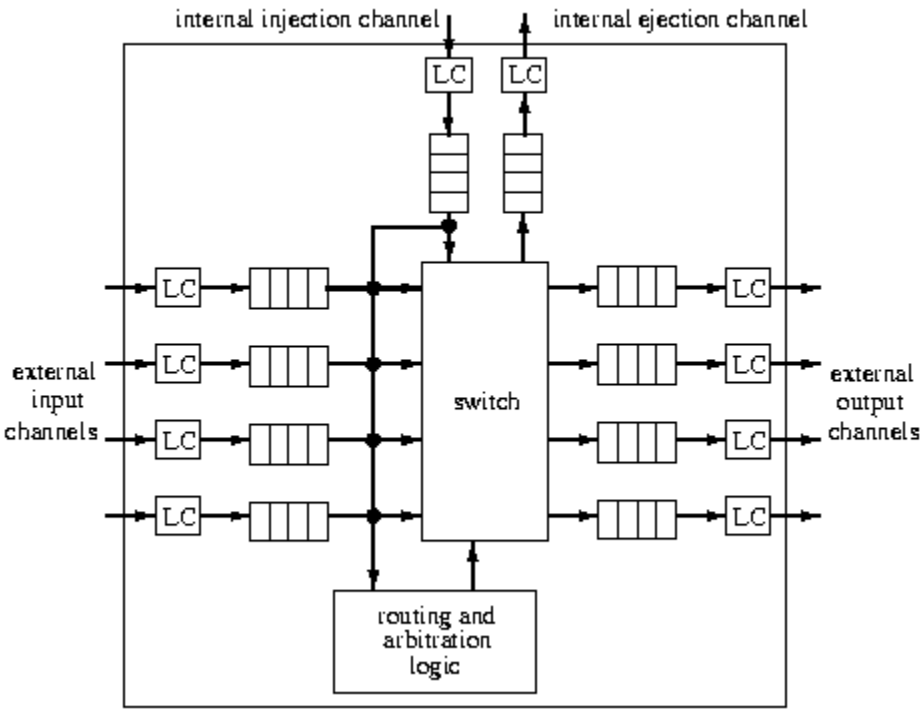


Figure 3: Architecture of Single-Port Router

Channel:

It comprises of communication medium, link controllers, and buffers.

Buffer:

FIFO(First In First Out) memory buffer for temporarily storing *one* or *several* **units of communication** in transit. They are required for storing transferred data until the next channel is reserved and is in a position to surpass them. A router can implement 3 types of buffering:

- **Output and input:** Here, every output and input external channel is connected to a single buffer.
- **Output:** Here, buffers are connected only to output external channels.
- **Input:** Here, buffers are connected only to input external channels.

2.2 COMMUNICATION UNITS

Message:

It is the communication unit from a programmer's point of view.

Packet:

These are the smallest communication unit containing sequencing information in its header part and routing information (such as a destination address). Its size has the order as hundreds or thousands of bytes or words. It consists of header flits, trailer flits and data flits.

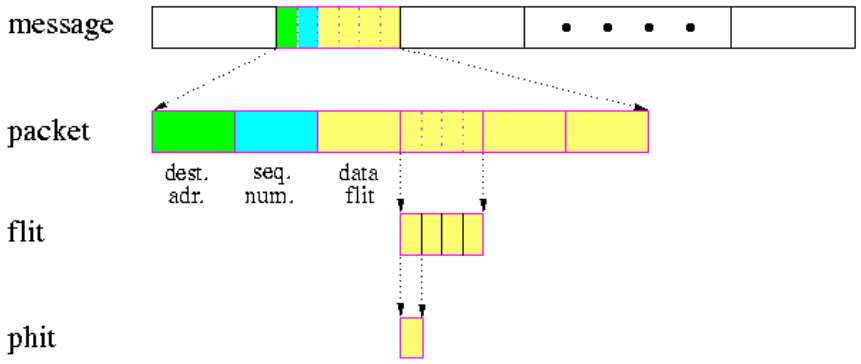


Figure 4: Hierarchy of Communication Units

Routing algorithm:

It calculates the pathway selected by a packet to reach its destination. It must decide within every intermediary router which output channel(s) should be chosen for incoming packets.

Routing and arbitration unit:

It executes the routing algorithm and packet flow control protocol and sets the switch accordingly e.g. resolving conflicts between concurrent requests for the same output link (using fixed channel priority , Round-robin, or first-come-first-serve policy).

Switch:

It connects output buffers with input buffers (channels). It may be a **crossbar** switch or some other substitute.

Switching mechanism:

It determines the manner in which network assets are to be allocated for data transmission. When and how the input channel should be connected to the output channel is decided by the routing algorithm. It is the real mechanism that takes data from the input channels and puts them on the output channels.

2.3 FLOW CONTROL

It defines the synchronization protocol used between sender and receiver nodes, which decide actions to be taken in case of fully filled buffers, deadlocks , faults, busy output links, etc. Flow control has no less than two levels:

- **Packet flow control:** It carries out synchronization establishment between receiver and sender at packet-level, guaranteeing availability of sufficient buffer space at the receiver end. It also controls allocation of buffers and channels to a packet as it travels through a routing pathway. It ensures that *resource collision* is tackled properly. A number of resource collision protocols are available. The packet that can't go further since the output channel chosen by the routing algorithm is busy, can be:
    - Stalled till enough buffer space becomes free. Till this happens, the resources that lay along the present pathway are stuck (wormhole switching),
    - stored in local router data buffers (store-and-forward, virtual cut-through switching),
    - Re-routed (adaptive routing),
    - Killed.
- **Physical channel flow control:** It implements the multi-cycle packet-flow control. It divides down the packets into flits. Even flits (flow digits) may take many cycles for transfer, thus the basic information unit is taken as phit.

Link controller:

It executes the physical channel flow control protocol between adjacent routers.

Flit:

It is the smallest unit of information at data link layer, and it has the size of one or a number of words. Flits might be of numerous types and flit transfer protocol normally needs many clock cycles.

Phit:

At the physical layer, it is the smallest information unit that is transferred via a single physical link in a single cycle.

2.4 NETWORK TOPOLOGY

Network Topology:

Mesh.

Mesh network topology comprises of m columns and n rows. The routers are located in the points of intersection of two wires and processing cores are placed close to routers. The address of routers and its resources can be characterized as (x,y) coordinates in a mesh.

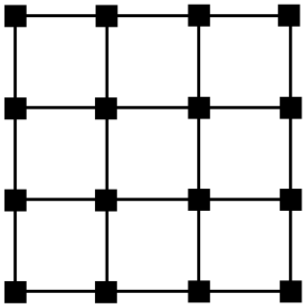


Figure 5: Mesh network

2.5 ROUTING ALGORITHMS

Routing decisions:

The most critical criteria for routing is *when* and *where* the routing calculation is done.

Source routing:

Source nodes determine complete routing path before inserting packets into the network so that intermediate routers only read the headers flits (and generally mark appropriate subfields in it) and mechanically set its switches as per demand. If output degree of a router is k, then header of a packet following a path of length d requires at least $d*log(k)$ bits for encoding d output channel numbers. In networks based on cartesian product, the size of the header routing information can be reduced by using **street-sign** routing. The default output channel is of the same direction and dimension as the input channel, unless the header starts with manually given new output channel number related with an address matching the address of the current router.

Implementing the routing algorithm:

Routing decisions must be fast. In distributed routing, a hardware implementation is advantageous. We have two basic approaches.

Finite-state machine:

Hardware/software algorithm that implements some finite-state automation.

Adaptivity of Routing algorithm:

Routing functions may be calculated based on address, but can also be based on other information.

1.6 ROUTING ALGORITHMS

2.6.1 Oblivious Routing Algorithms

These have zero information about network situation, such as amount of traffic or congestion. A router takes routing decisions randomly or on the basis of some algorithm. *Minimal turn* routing the simplest example of oblivious routing algorithm. It routes packets by using minimum turns possible.

2.6.1.1 Dimension Order Routing

An example of minimal turn algorithm is Dimension order routing (DOR). This algorithm determines the direction in which packets are routed at each stage of routing.

2.6.1.1.1 XY routing

XY routing is an example of dimension order routing that sends packets first in x-direction to the required column and then in y-direction to the receiver node. XY routing works well on a network having torus or mesh topology. Addresses of routers are their x and y coordinates. This type of routing never finds itself in a livelock or deadlock.

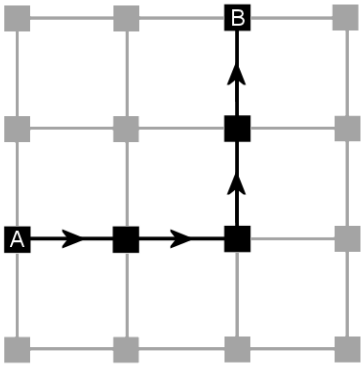


Figure 6: XY routing from router A to router B.

Traditional XY routing faces some problems. The traffic doesn't spread evenly over the entire network since this algorithm causes maximum load in the centre of network. We require algorithms which evenly divide the traffic load over the entire network. （1）

2.7 SWITCHING

Wormhole (WH) switching

- The incoming packets are broken down into flits which are passed through the path in the exact pipelined manner as in conflict-less virtual cut-through switching. So, in this case, transmitting of separate packets can't be multiplexed or shared over a single physical channel devoid of extra architectural support.

- The main difference between wormhole and virtual cut-through is that the routers don't have buffers for the entire packet. Rather, each router possesses *small* buffers for a single or few flits.
- The header flit constructs a pathway in the network, which is followed by remaining flits in a pipeline manner. The series of links and buffers occupied by flits belonging to a specific packet constitutes what is known as the **wormhole**. The length of the pathway is directly proportional to the number of flits present in the packet. Generally, it covers the whole path connecting destination and source routers.
- If the header flit can't go further due to blocked output channels, entire chain of flits is *stopped*, and they occupy the flit buffers in the routers lying on the path constructed till then and block any other possible transmission.

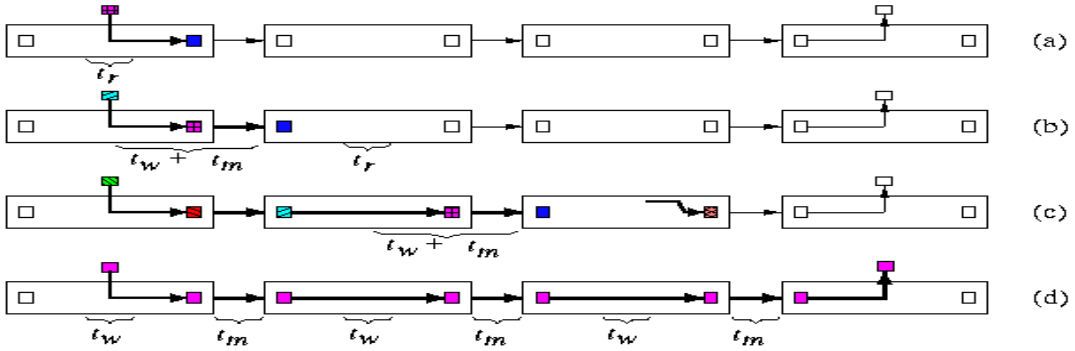


Figure 7: Wormhole switching of a packet

(a) After the routing function is calculated, the header flit is copied into the output buffer.

(b) The header flit is passed on to the second router. Other flits follow it.

(c) When the header flit arrives in a router with busy output channels, the entire chain of flits present along the path get stopped, leading to blockage of all its channels.

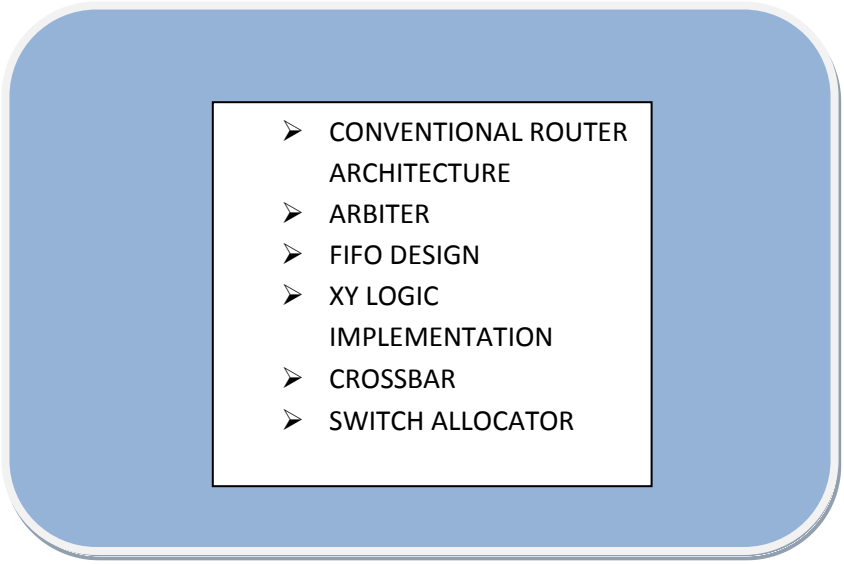
(d) Pipelining of flits in the ideal case of conflict-free routing, implementing the concept of wormhole switching across all routers.

Remarks

- Wormhole routing permits fast, cheap, small and simple routers. Thus, this is the most commonly used switching technique of today.

- The biggest drawback of this technique is blocked resources in case of stalled pipelines.
- Wormhole switching is very **deadlock-prone**.
- This problem is associated with the idea of virtual channels (VC). as flits don't carry any identity flags, we cannot mix flits of different packets into a single buffer. Still, we can divide a single physical channel into several virtual channels by adding some control logic. They will possess separate buffers, but will multiplex over the single physical channel.
- Routers implementing Wormhole switching often use input buffering only.

3. CONVENTIONAL ROUTER ARCHITECTURE



3.1 CONVENTIONAL ROUTER ARCHITECTURE

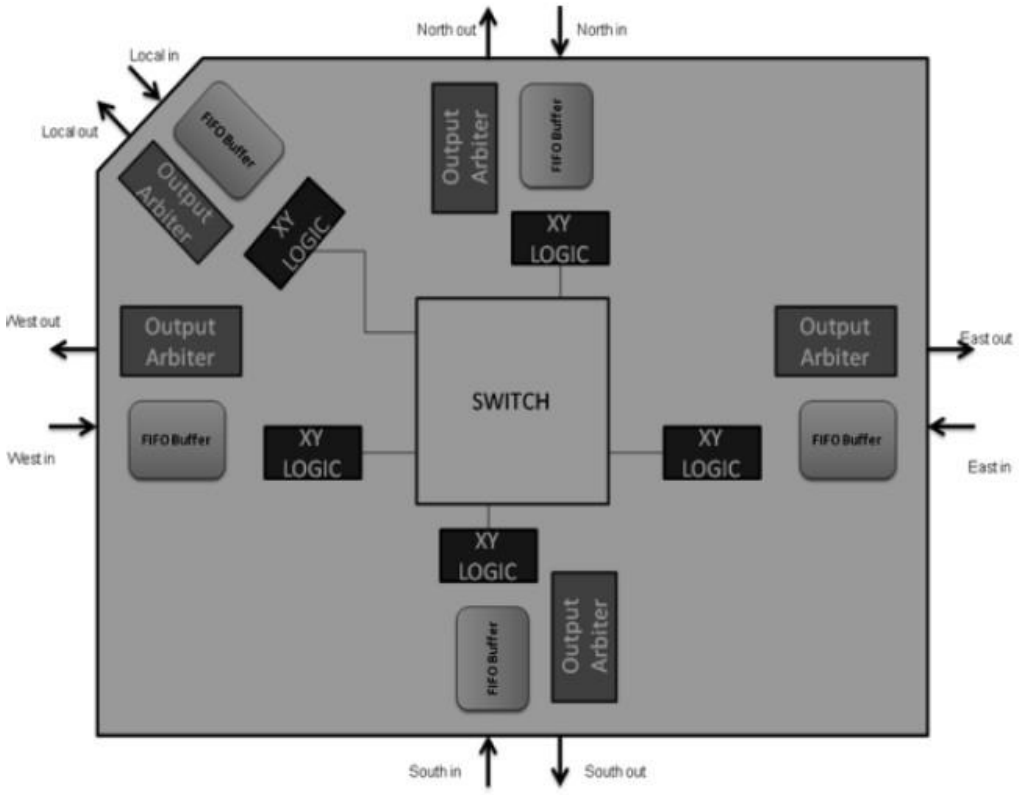


Figure 8: Conventional Router Architecture

In the conventional router architecture, each router has five ports, one for each direction: North, South, East and West, and a local port that is connected to the local processing element. Each port has one output arbiter, one FIFO Buffer and one XY logic implementation block. Finally, all ports are connected to each other using Switch allocator, which contains crossbar switch for depicting physical connections.

3.2 ARBITER DESIGN

Use of an arbiter in an output channel is to tackle the problem of a single port receiving multiple input requests. Arbiter matches *N requests to 1 resource*. Arbiter used here is based on the rotating priority scheme where each port's priority reduces once it is served.

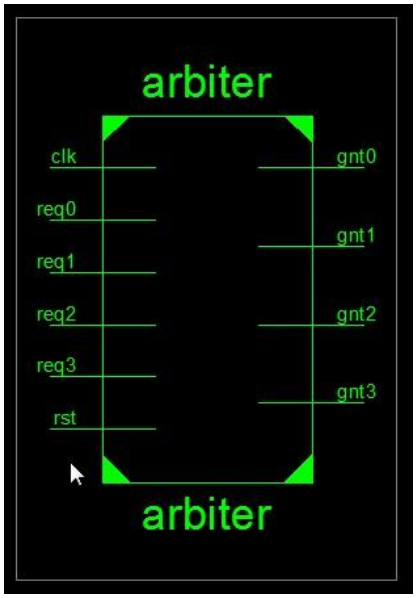


Figure 9: RTL Schematic of Arbiter

Figure 10: Testbench for Arbiter

3.3 FIFO BUFFER DESIGN

The FIFO memory buffer is used as a memory buffer to store the incoming data temporarily. It is used for synchronizing the speeds of operation of the receiver and sender. The receiver and sender have different frequencies, and if sender sends at a higher frequency compared to receiver, receiver will get swamped with incoming data. In order to avoid this, and also for temporary storage of incoming data, we use FIFO (first-in-first-out) memory buffers.

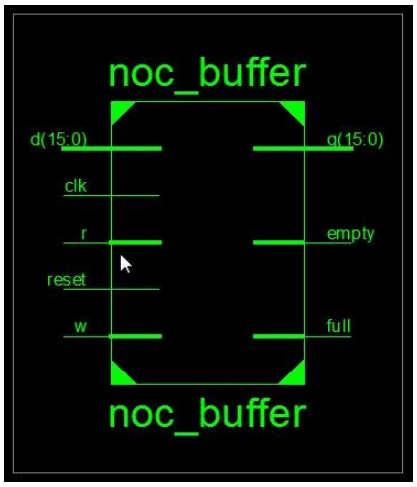


Figure 11: RTL Schematic of FIFO Buffer

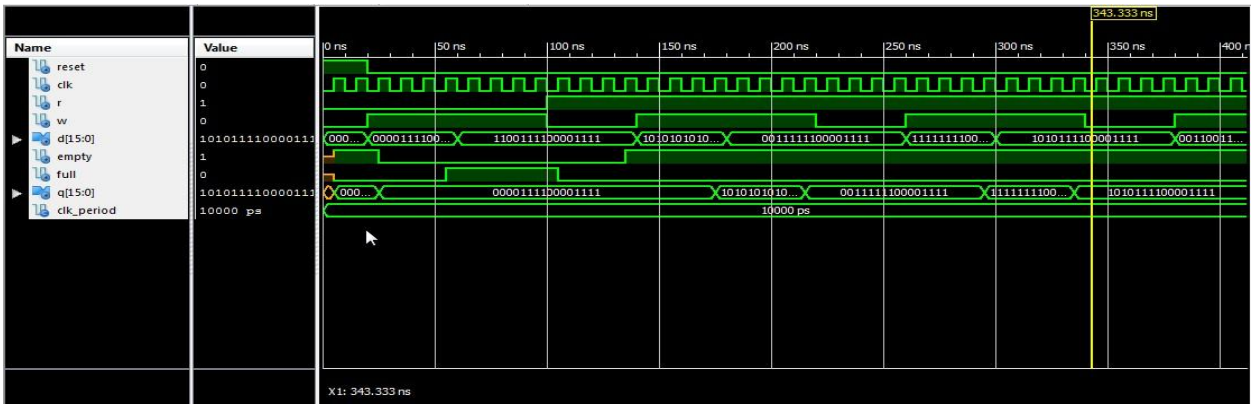


Figure 12: Testbench of FIFO Buffer

3.4 XY LOGIC IMPLEMENTATION

XY Logic Block carries out comparison of the coordinates present in header flit with the coordinates stored locally and hence estimates the direction in which the packet needs to be switched. The router has a switch that switches data from the input port of a specific direction of router to the output port of the required coordinate calculated by the XY routing algorithm. According to the XY routing algorithm, the data flit traverses first in the x-direction to reach the correct x-coordinate, then it goes in the y-direction to reach the destination.

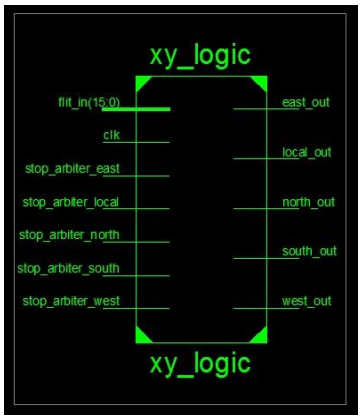


Figure 13: RTL Schematic of XY Logic Implementation Block

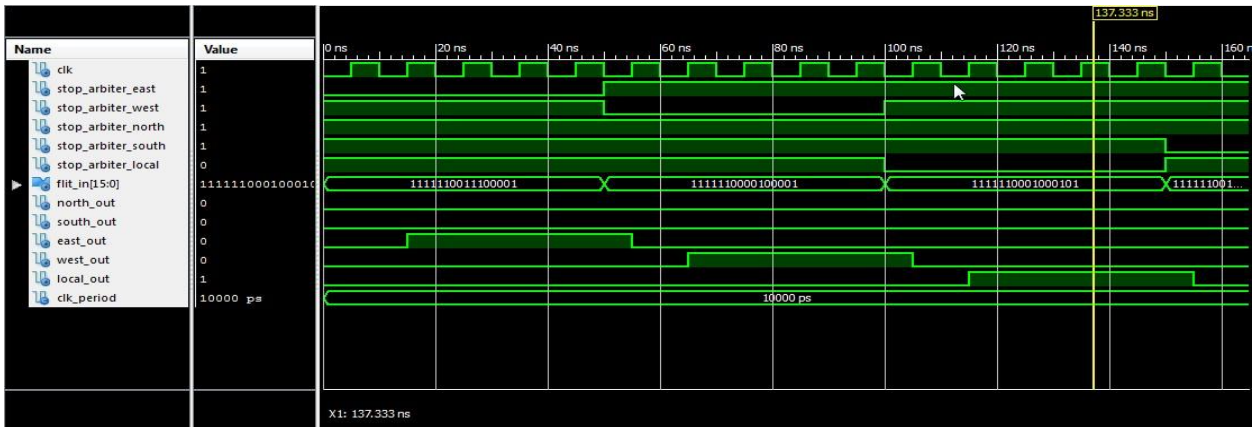


Figure 14: Testbench of XY Logic Implementation Block

3.5 CROSSBAR

A crossbar depicts the physical connections inside the switch allocator. It is a collection of switches arranged in a matrix configuration. The crossbar switch has multiple input and output lines that form a crossed pattern of interconnecting lines between which a connection may be established by closing a switch located at each intersection, the elements of the matrix. It switches bits from input ports to output ports, performing the essence of a router's function.

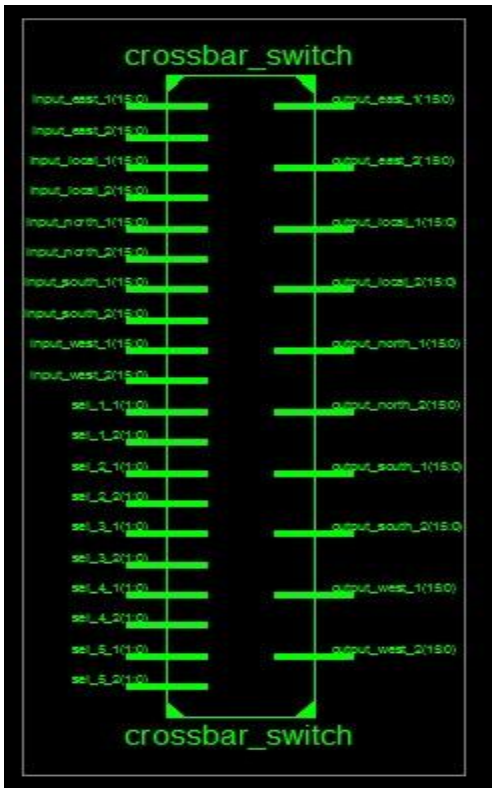


Figure 15: RTL Schematic of Crossbar Switch

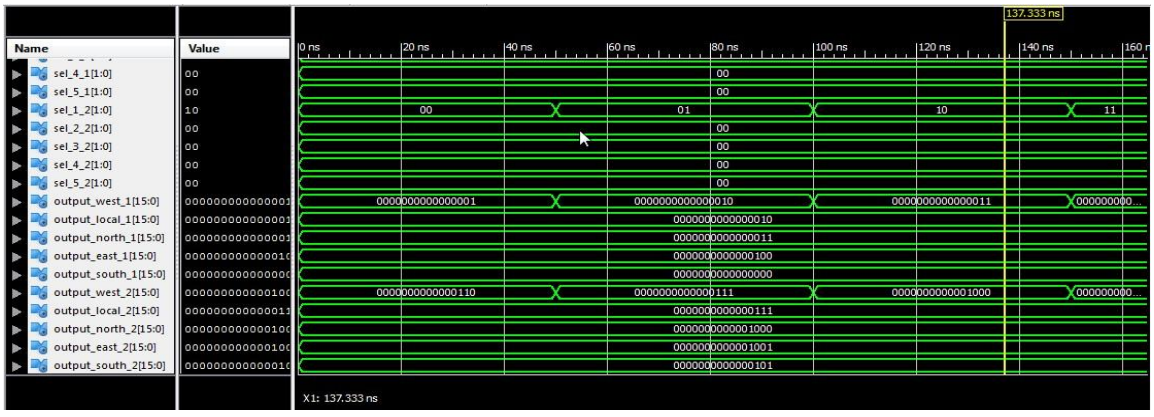


Figure 16: Testbench of Crossbar Switch

1.7 SWITCH ALLOCATOR

It controls switching of data from one port to other. Allocator consists of many arbiters. These arbiters map between input and output paths. Switch allocator contains the crossbar switch which depicts the physical connections.

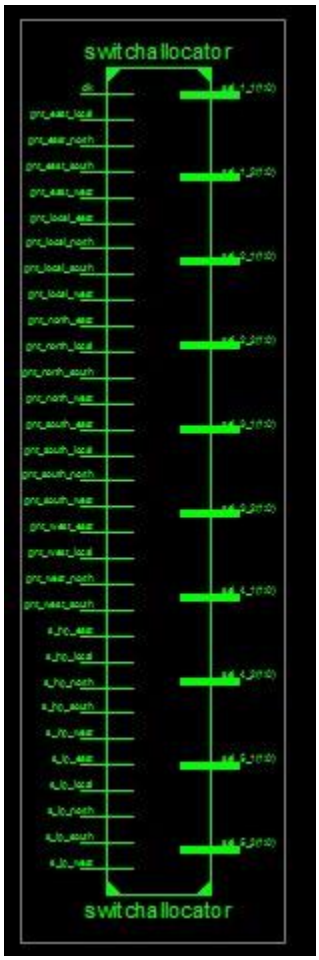


Figure 17: RTL Schematic of Switch Allocator

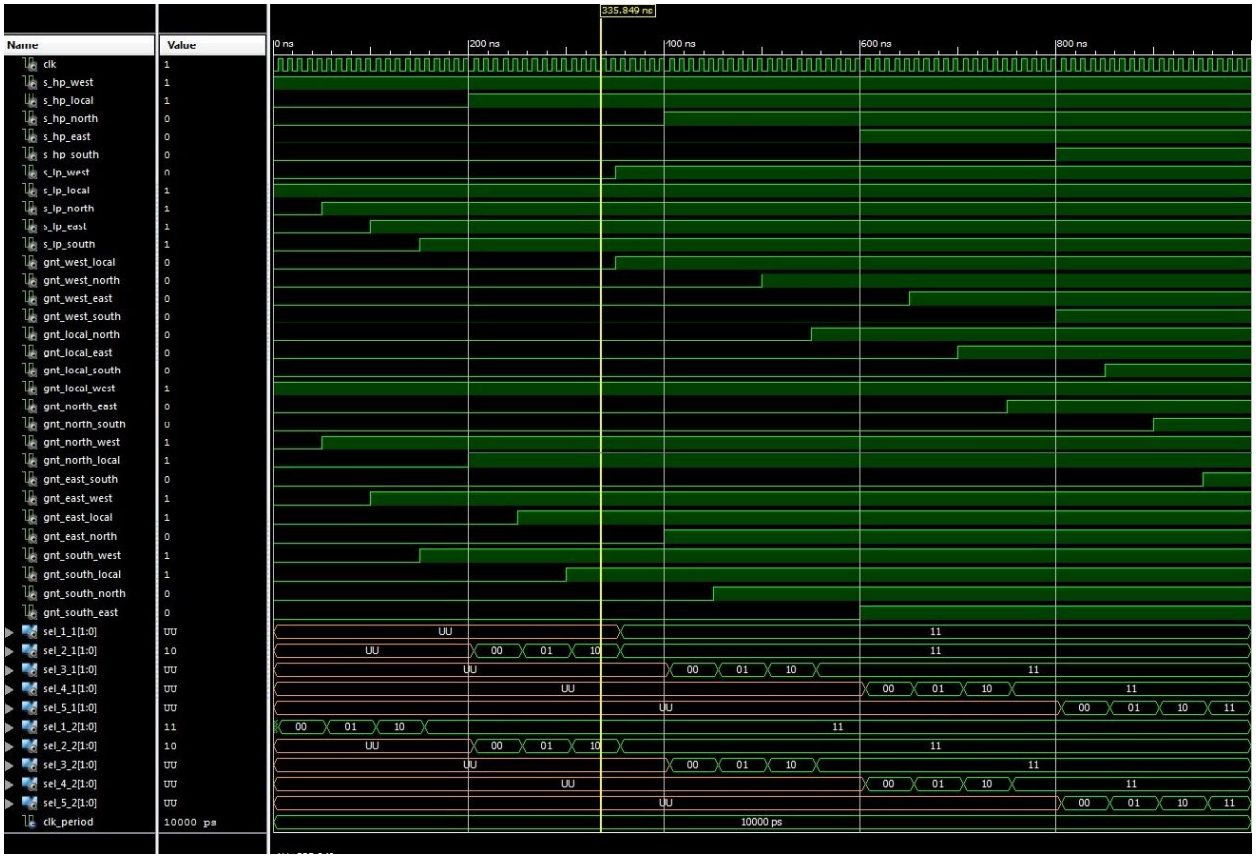
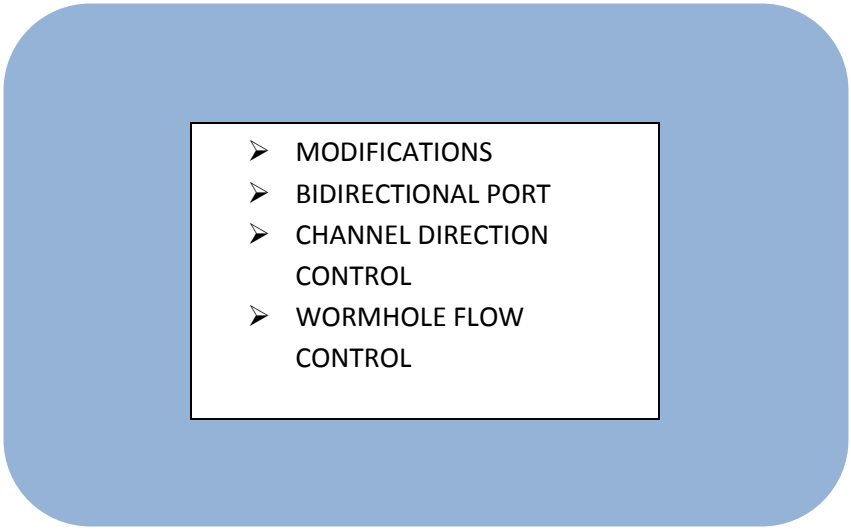


Figure 18: Testbench of Switch Allocator

4. BINOC ROUTER ARCHITECTURE



4.1 BINOC ROUTER ARCHITECTURE

Modifications done to the 5-port router structure when used in BiNoC are:

1. Every port of the router has to be a bidirectional port.

2. Every port should have buffers.

3. A low priority finite state machine (LP FSM) and a high priority finite state machine (HP FSM) should be connected at every port so that dynamic reconfiguration of the channel direction as per traffic needs is possible.

4. To tackle heavy traffic, increase the FIFO buffer's capacity.

4.2 MODIFIED BINOC ARCHITECTURE

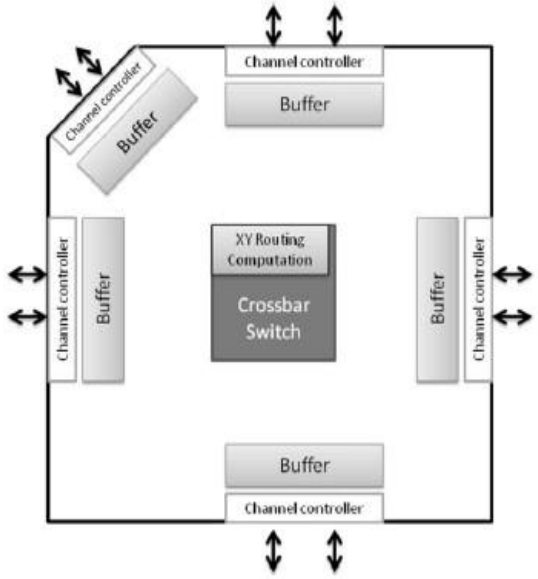


Figure 19: Modified BiNoC Architecture

In the modified BiNoC architecture, the aforementioned modifications are carried out.

4.3 BIDIRECTIONAL PORT

Now, each port is made bidirectional, meaning each channel connected to each port has the capability of both sending and receiving data.

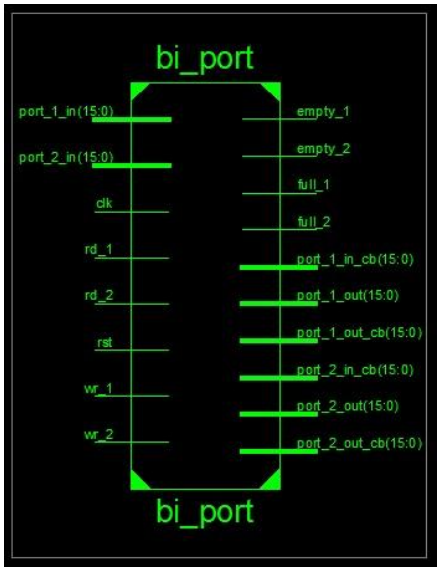


Figure 20: RTL Schematic of Bidirectional Port

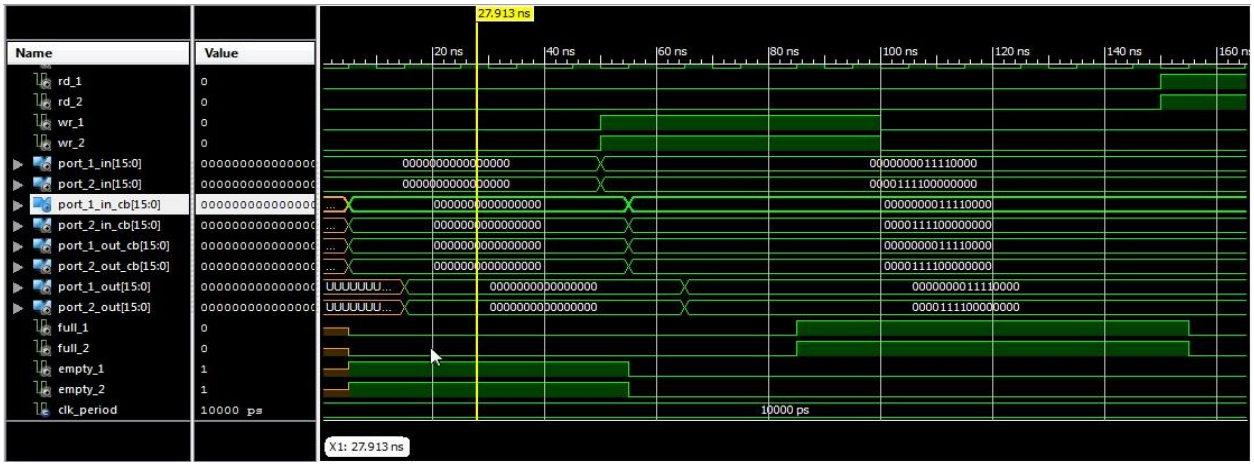


Figure 21: Testbench of Bidirectional Port

4.4 CHANNEL DIRECTION CONTROL

- In a bidirectional channel, the direction for data transmission has to configure by itself, based on traffic demands, that too at run time. For this purpose, a distributed Channel direction Control (CDC) protocol is utilized.
- This direction reconfiguration is controlled by a pair of FSM(Finite State Machine)s that are present in the routers at both ends in their channel control blocks.
- Opposite priorities (high,low) are given to FSMs present on the complementary channel of the same pair of neighboring routers.
- Each FSM undergoes a sequence of three states: Idle, Free, and Wait, defined as:

1. **Idle State:** Here, the channel is available to receive data from the neighboring router.
2. **Free State:** Here, the channel is ready for sending data to the neighboring router.
3. **Wait State:** It is an intermediary state meant for preparing the transition from the Idle state with an input channel direction to the Free State with an output channel direction.

- These two Finite State Machines exchange control signals via some hand-shaking signals: named as output request (output_req) and input request (input_req).

- The output_req is made “1” when the sending end router has a data packet to transmit. The output_req signal from one router serves as the input_req signal to the FSM of other router. When there is a grant given to a particular port to output data, each FSM also receives a channel request (channel_req) signal from the internal routing calculation module. When a data packet in the local router is requesting the current channel, channel_req = 1.

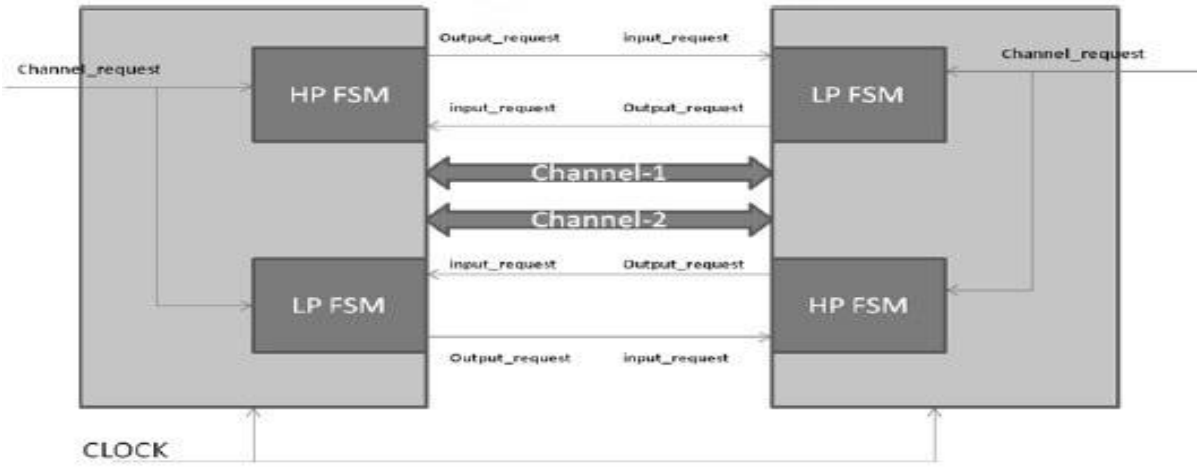


Figure 22: Channel Control Blocks of Two Adjacent Routers

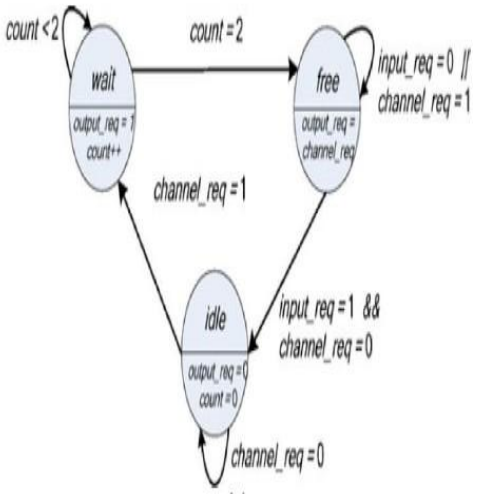


Figure 23: State Diagram of High Priority FSM

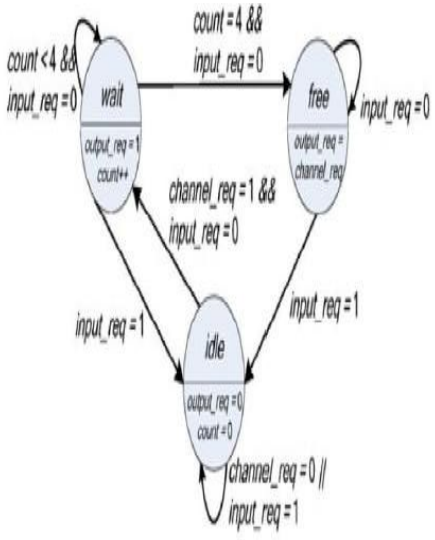


Figure 24: State Diagram of Low Priority FSM

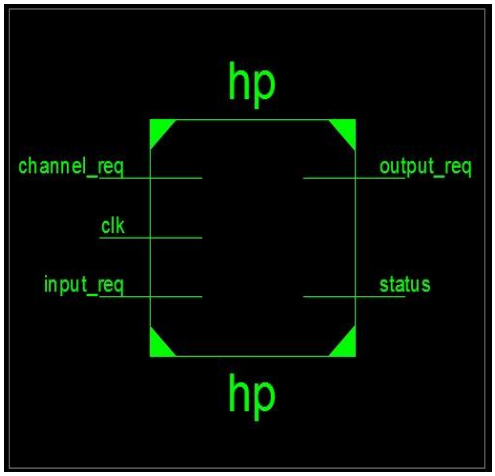


Figure 25: RTL Schematic of High Priority FSM

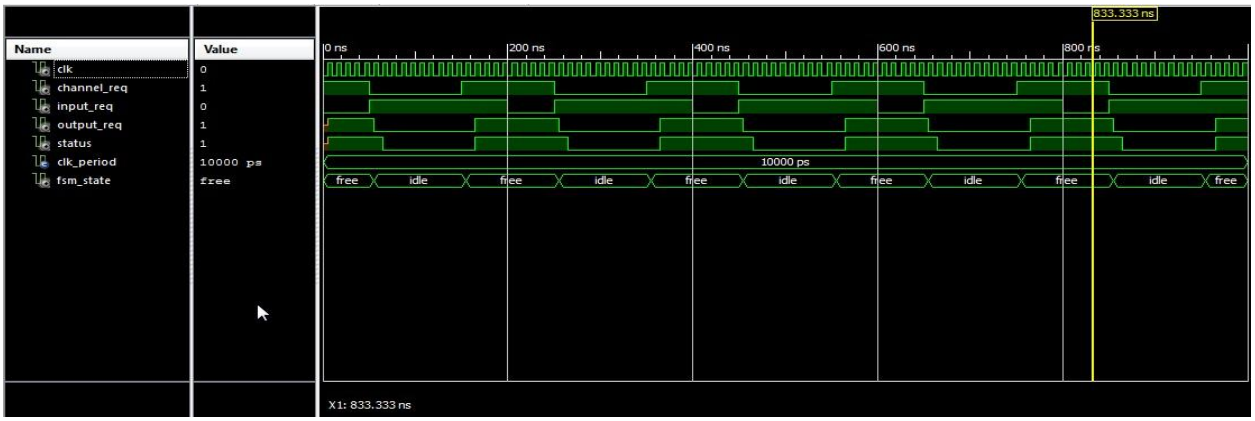


Figure 26: Testbench of High Priority FSM

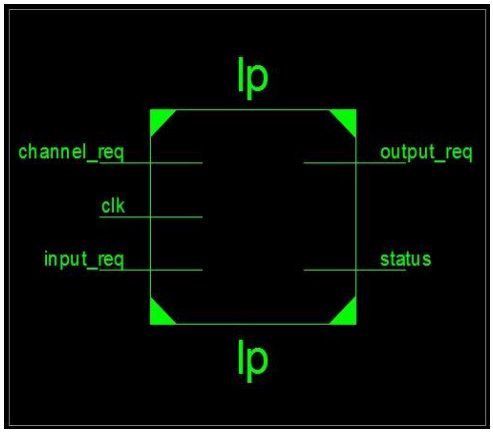


Figure 27: RTL Schematic of Low Priority FSM

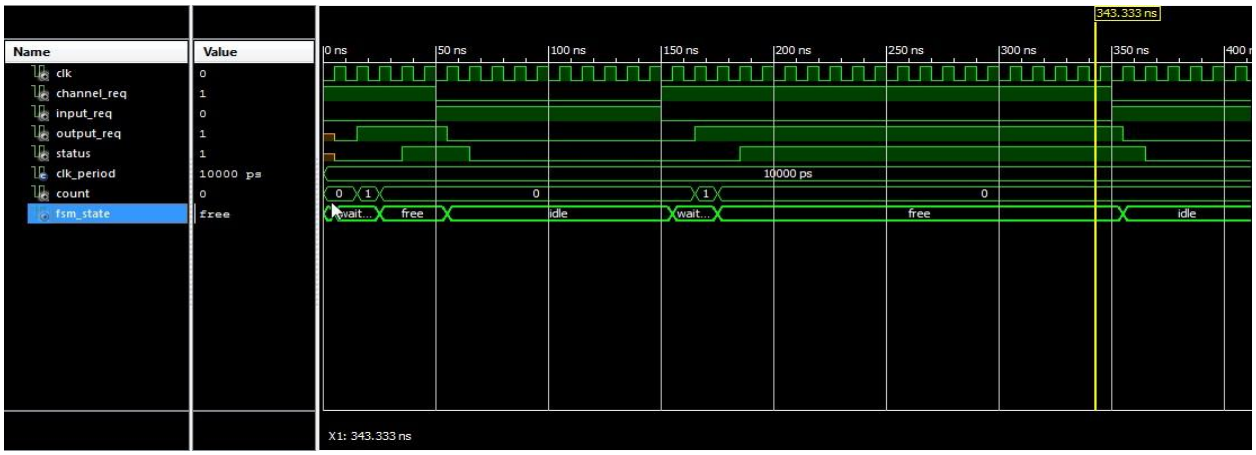


Figure 28: Testbench of Low Priority FSM

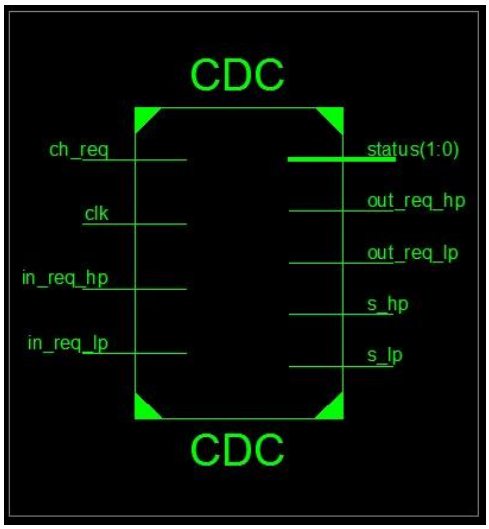


Figure 29: RTL Schematic of Channel Direction Control Block

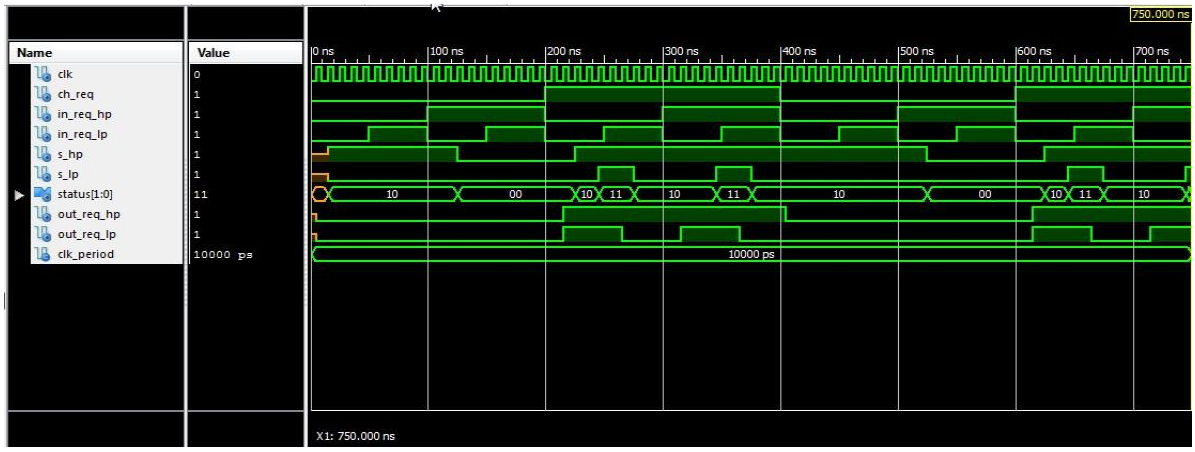


Figure 30: Testbench of Channel Direction Control Block

4.5 WORMHOLE FLOW CONTROL

This flow control has merits such as less memory requirement and less latency. In this type of routing, incoming packets are broken down into small and equal-sized flits (short for flow control digits). After first flit passes through a route, the route is set aside to route the remaining flits of the packet. Wormhole is the name given to this route.

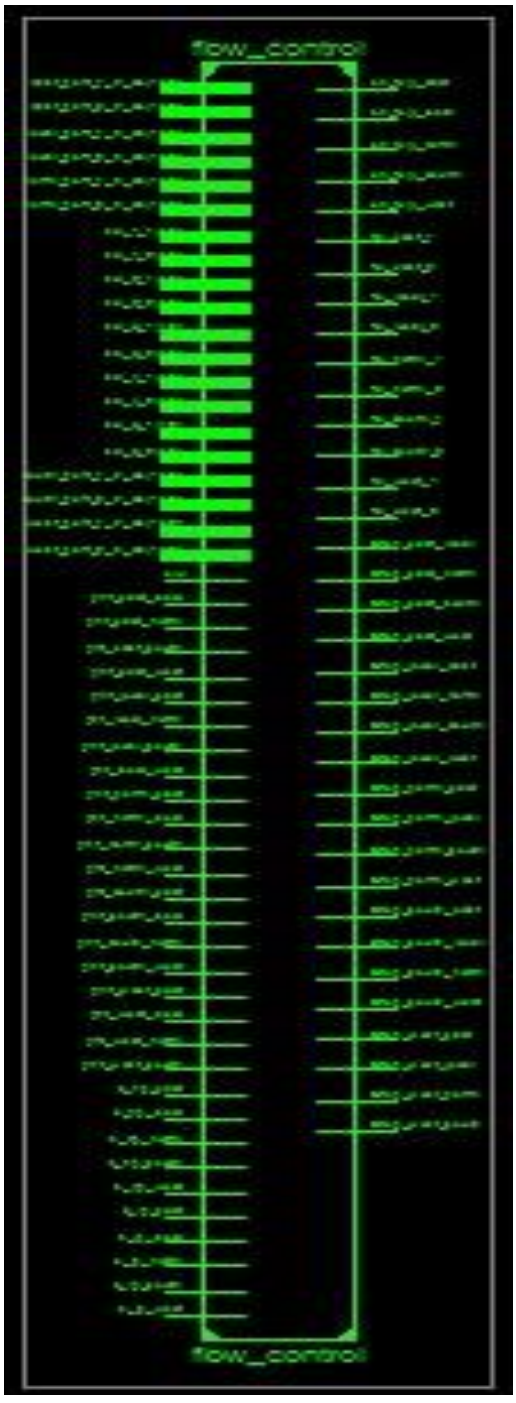
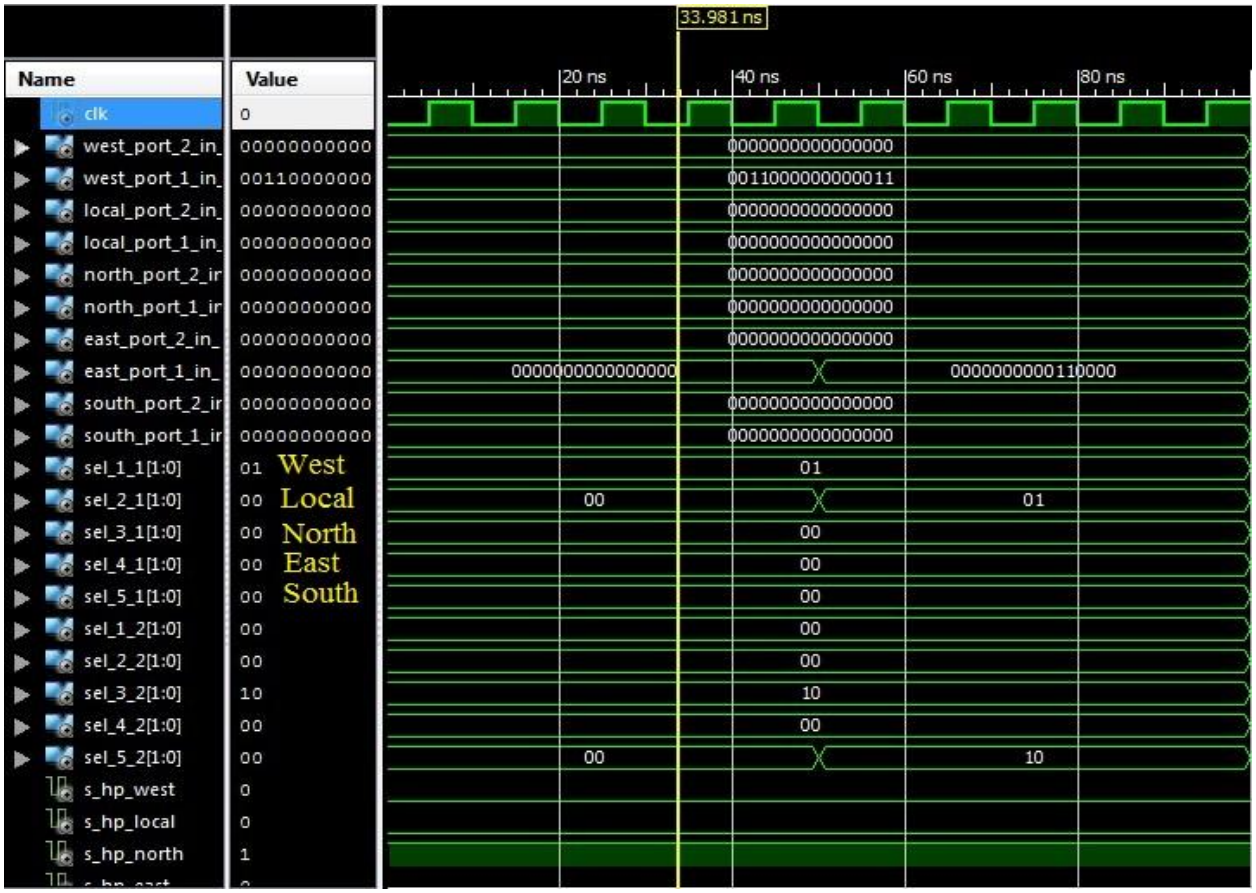
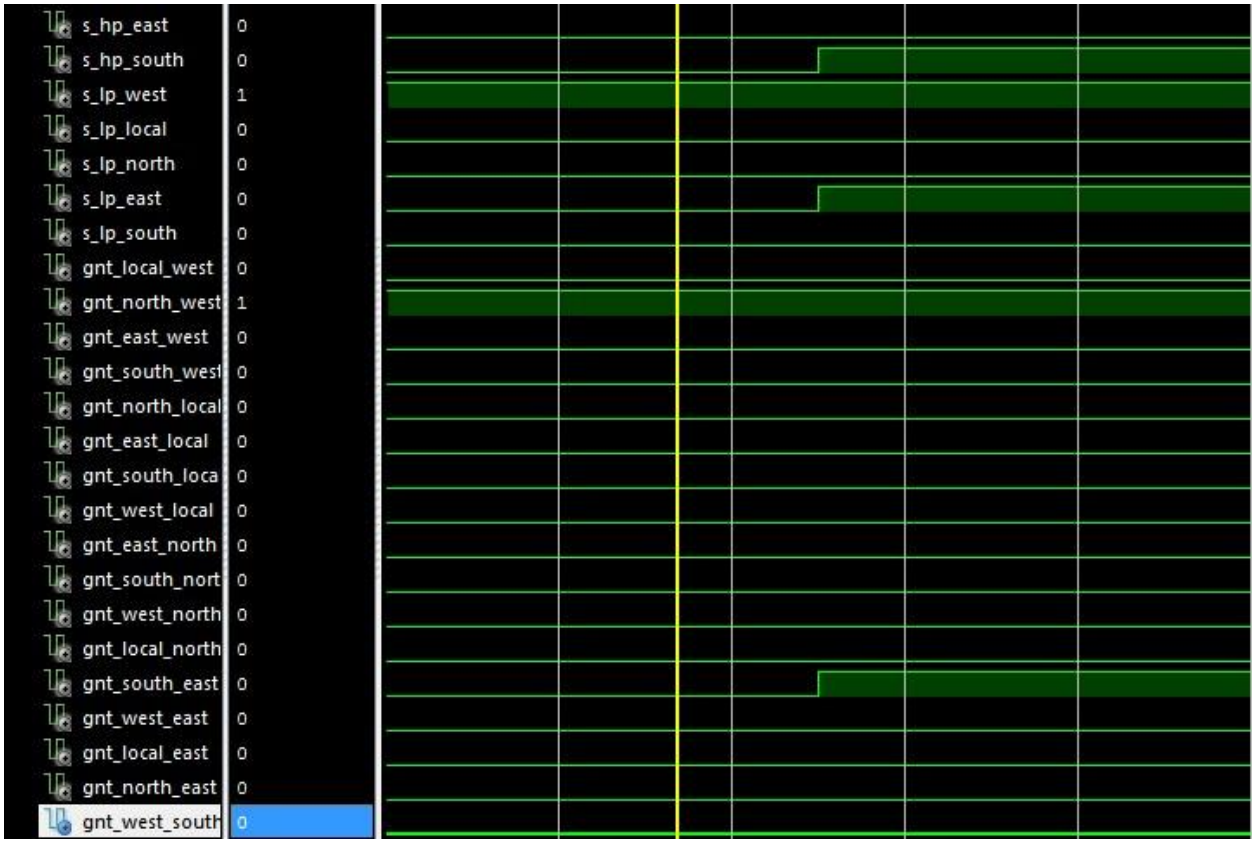


Figure 31: RTL Schematic of Flow Control Block


33.981 ns
20 ns
40 ns
60 ns
80 ns
Name
Value
clk
west_port_2_in_
west_port_1_in_
local_port_2_in_
local_port_1_in_
north_port_2_ir
north_port_1_ir
east_port_2_in_
east_port_1_in_
south_port_2_ir
south_port_1_ir
sel_1_1[1:0]
sel_2_1[1:0]
sel_3_1[1:0]
sel_4_1[1:0]
sel_5_1[1:0]
sel_1_2[1:0]
sel_2_2[1:0]
sel_3_2[1:0]
sel_4_2[1:0]
sel_5_2[1:0]
s_hp_west
s_hp_local
s_hp_north
West
Local
North
East
South


s_hp_east
s_hp_south
s_lp_west
s_lp_local
s_lp_north
s_lp_east
s_lp_south
gnt_local_west
gnt_north_west
gnt_east_west
gnt_south_west
gnt_north_local
gnt_east_local
gnt_south_local
gnt_west_local
gnt_east_north
gnt_south_north
gnt_west_north
gnt_local_north
gnt_south_east
gnt_west_east
gnt_local_east
gnt_north_east
gnt_west_south

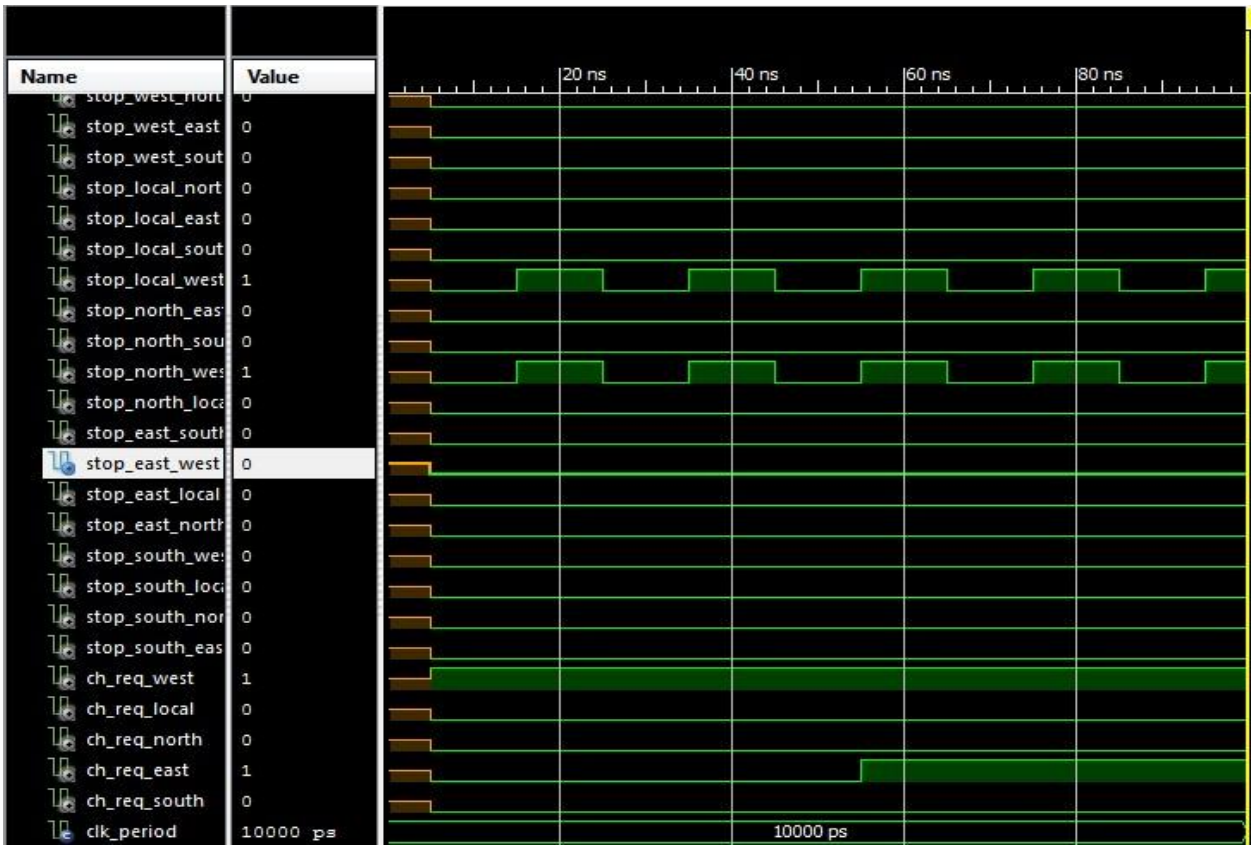


Figure 32: Testbench of Flow Control Block

Suppose, we give higher priority to the port from which output has to be read. So the intermediate port is given lower priority. From the testbench above, we can see that channel allocation is done as required.

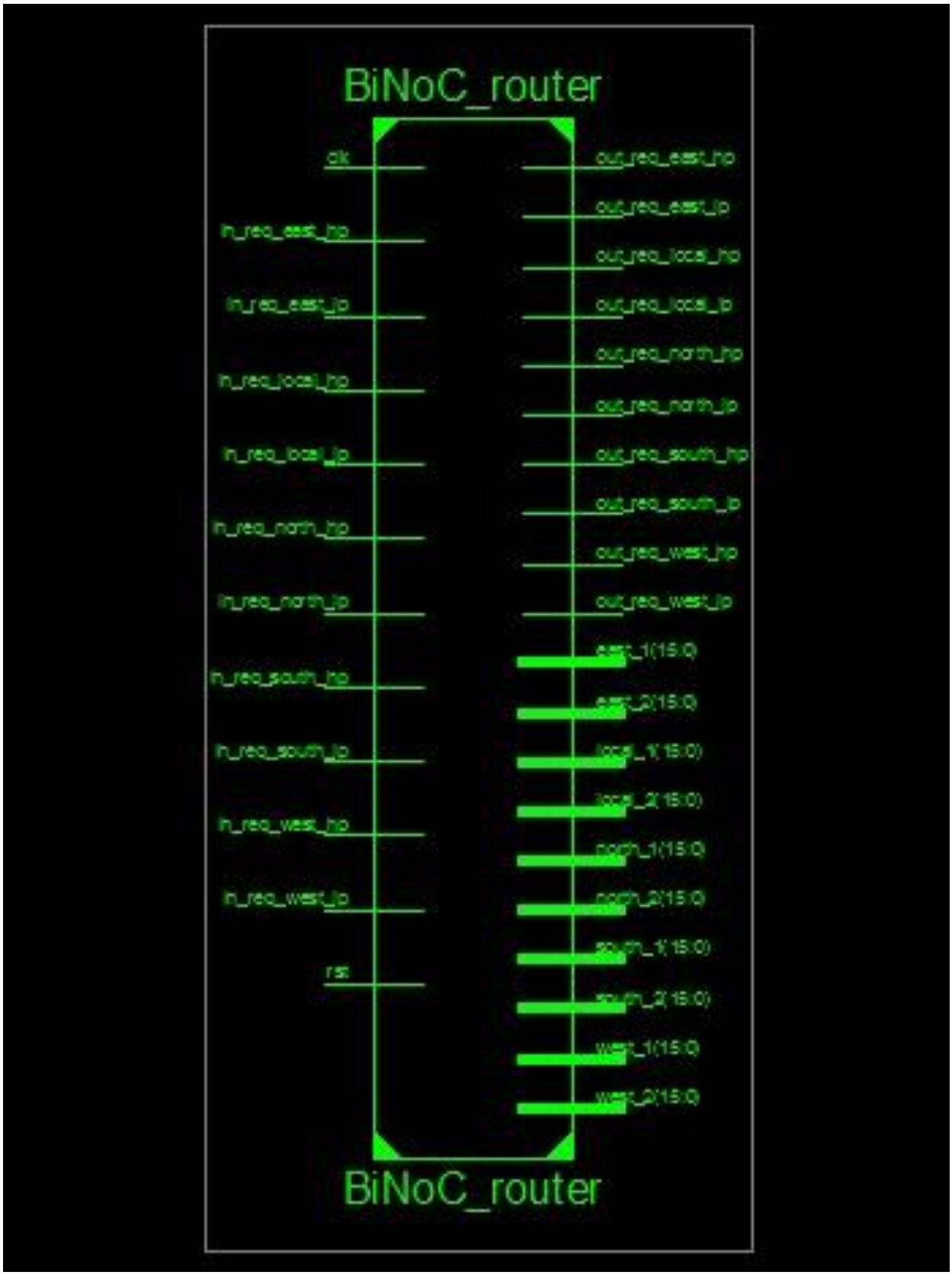
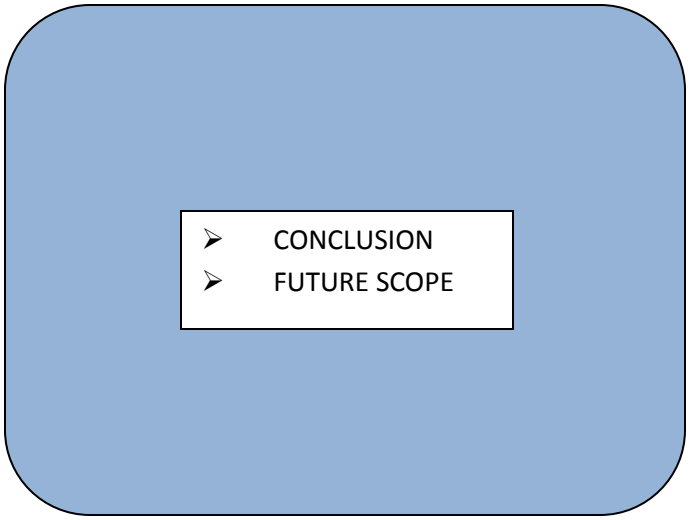


Figure 33: RTL Schematic of BiNoC Router

5. CONCLUSION AND FUTURE SCOPE



5.1 CONCLUSION

Thus, a bidirectional router for Network-on-chip was designed, which used Channel Direction Control protocol and Wormhole switching. This is much more efficient compared to other conventional routers. It leads to better resource utilization as well.

5.2 FUTURE SCOPE

- Use of Error Detection and Retransmission schemes for Error Control
- Integration of all modules
- Implementation on FPGA

6. REFERENCES

6.1 REFERENCES

1. [Online]http://pages.cs.wisc.edu/~tvrdik/7/html/Section7.html.
2. *A broad strategy to detect crosstalk faults in network-on-chip interconnects.* **M.BotelhoF.L.Kastenmidt,M.Lubaszewski,E.Cota,L.Carro**2010. 18th IEEE/IFIP International Conference on VLSI and System-on-chip. pp.298-303.
3. *An infrastructure IP for online testing of networkon- chip based SoCs.* **P.BhojwaniR.N.Mahapatra**2007. 8th International Symposim on Quality Electronic Design, ISQED'07. pp.867-872.
4. *Asymptotic Analysis of Dynamic Algorithms Designed to Provide Parallel Communication among NoC in NiP using MIN.* **NitinNitin,Rahul Mehta, Palak Sethi, Esha Gupta, Rohit Sharma, and Vivek Kumar Sehgal**2009. 2009 11th International Conference on Computer Modelling and Simulation.
5. *GARNET: A detailed on-chip network model inside a full-system simulator.* **Niket AgarwalTusharKrishna,Li-Shiuan Peh, Niraj K. Jha**April 2009. IEEE International Symposium on Performance Analysisof Systems and Software. pp.33-42.
6. IJMER. [Online]www.ijmer.com.
7. *Input buffer planning for network-on-chip.* **Yaming Yin,Shuming Chen, and Xiao Hu**2010. 2010 International Conference on Computer application and System Modelling.
8. **Li-Shiuan PehNatalieEnright Jerger**On-chip Networks. *Synthesis Lectures on Computer Architecture.* s.l., Morgan & Claypool publishers, 2009. （ISBN: 9781598295856）.
9. Nitrkl ethesis. [Online]ethesis.nitrkl.ac.in.
10. *Noise Tolerant and Faster On Chip Communication Using Binoc Model.* **Arun ArjunanKarthikaManilal**5, Oct 2013, Vol. 3. （ISSN 2249-6645）.
11. studymode. [Online]www.studymode.com.
12. techylib. [Online]www.techylib.com.